
Semestrální práce IUJC 2020

Způsob vypracování a odevzdání práce:

- Semestrální práce je vypracována samostatně (není přípustná **duplicita** kódu)
- Student může být v případě potřeby vyzván k obhájení práce, tzn., že student musí být schopen zodpovědět dotazy k práci
- Zápis kódu odpovídá zvykům strukturovanému programování
- Všichni studenti odevzdávají práci do STAGU v sekci odevzdávání prací (Blok: Sem IUJC 2020 - Semestrální práce IUJC 2020)
- **Nejzazší** termín odevzdání práce je **29. 12. 2020 do 20:00** (po tomto termínu nebudou semestrální práce přijímány a student ztrácí možnost získání zápočtu)
- Odevzdávají se pouze **zdrojové** soubory (*.c, *.h), archivovány ve **ZIPu** (ne v .rar, .tar, atd)
- Použít funkci `_CrtDumpMemoryLeaks()` pro ověření, že veškeré alokované paměťové prostředky byly navráceny zpět.

Agenda personální agentury

Hlavní idea

Personální agentura potřebuje pro svou efektivní činnost SW nástroj pro správu její agentury. Za tímto účelem udržuje dva základní seznamy (**lineární seznam**). První udržuje seznam **kandidátů** poptávající práci, druhý pak jednotlivé nabízené pracovní **pozice** firem. Data pro tyto seznamy je možné importovat ze souboru CSV. Pokud personální agentura někoho doporučí na **pohovor** eviduje ho v dynamickém poli pohovorů.

Struktura programu

Hlavičkový soubor `enums.h` obsahuje

```
enum OBOR
{Administrativa,Ekonomie,Pravo,IT_All,Zdravotnictvi,Obchod,Vyroba,Stavebnictvi,Skolstvi,
Doprava,Management};

enum KRAJ
{Praha,Stredocesky,Jihocesky,Plzensky,Karlovarsky,Ustecky,Liberecky,Kralovehradecky,Pardubicky,Vys
ocina,Jihomoravsky,Olomoucky,Zlinsky,Moravskoslezsky};

enum STAV_POHOVORU
{nenastaveno,nastaveno,zaslano_CV,prijat,neprijat,odmitl,pozastaven};
```

Modul kandidát

Hlavičkový soubor `kandidat.h` obsahuje

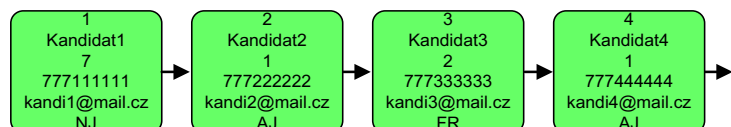
```
typedef struct kandidat {
    int id;
    char jmeno[50];
    enum OBOR obor;
    char tel[50];
    char mail[50];
    char jazyky[50];
    struct kandidat *dalsi;
} stKandidat;
```

Deklaraci funkcí:

```
stKandidat * vytvorKandidata(int id, char* jmeno, enum OBOR obor, char* tel, char* mail, char*
jazyky) // vraci adresu dynamické alokace kandidáta, ukazatel dalsi se nastaví na NULL
```

```
void vypisKandidata(stKandidat * kandidat) //vypíše kandidáta na obrazovku
```

soubor `kandiat.c` obsahuje **pouze implementace** (definici) těchto funkcí



Modul pozice

Hlavičkový soubor *pozice.h* obsahuje

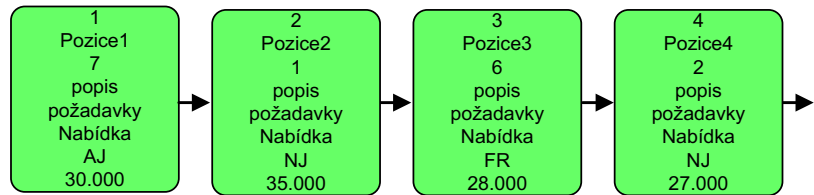
```
typedef struct pozice {  
    int id;  
    char pozice[100];  
    char popis[50];  
    char pozadavky[50];  
    char nabídka[50];  
    char jazyky[50];  
    float maxPlat;  
    enum KRAJ kraj;  
    struct pozice *dalsi;  
}  
stPozice;
```

Deklaraci funkcí:

```
stPozice * vytvorPozici(int id, char* pozice, enum KRAJ kraj, char* popis, char* pozadavky, char*  
nabidka, char* jazyky, float maxPlat) // vrací adresu dynamické alokace pozice, ukazatel dalsi se  
nastaví na NULL
```

```
void vypisPozici(stPozice* pozice) //vypíše pozice na obrazovku
```

soubor *pozice.c* obsahuje pouze implementace (definici) těchto funkcí



Modul pohovor

Hlavičkový soubor *pohovor.h* obsahuje

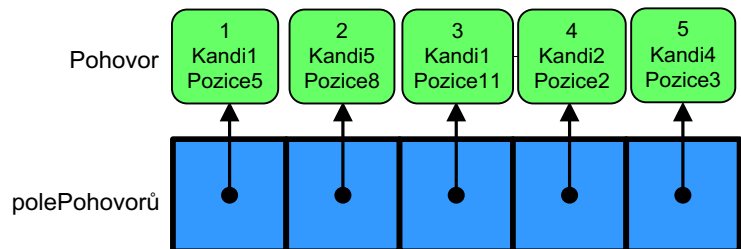
```
typedef struct pohovor {  
    int id;  
    stKandidat *kandidat;  
    stPozice * pozice;  
    enum VYSLEDEK_POHOVORU vysledek;  
}  
stPohovor;
```

Deklaraci funkcí:

```
stPohovor *vytvorPohovor(stKandidat *kandidat, stPozice *idPozice) // vrací adresu dynamické  
alokace pozice, stav je nastaven na 'nenastaveno', id pohovoru roste automaticky (1,2,..
```

```
void vypisPohovor(stPohovor * pohovor)//vypíše pohovor(včetně kandidáta a pozice)
```

soubor *pohovor.c* obsahuje pouze implementace (definici) těchto funkcí



Modul Agenda

agenda.h

deklarace funkcí pro práci se seznamy

```
void nactiSeznamKandidatu (char* nazevSouboru) // postupně načte jednotlivé kandidáty ze souboru  
a popřídává je za sebe do lineárního seznamu
```

```
void nactiSeznamPozic (char* nazevSouboru) // postupně načte jednotlivé pozice ze souboru a  
popřídává je za sebe do lineárního seznamu
```

```
void vypisSeznamKandidatu() // vypíše kandidáty se seznamu na každý řádek jednoho (používá  
opakovaně funkci pro výpis jednoho kandidáta)
```

```
void pridejKandidata(stKandidat * kandidat)// přidá kandidáta na konec lineárního seznamu
```

```

stKandidat *odeberKandidataZeSeznamu(int cisloKandidata) // odebere kandidata z lin. seznamu podle
jeho čísla, vrátí ho, a následně dealokuje

stKandidat *najdiKandidataZeSeznamu(int cisloKandidata) // najde kandidata z lin. seznamu podle
jeho čísla

void zrusSeznamKandidatu() // dealokuje celý lin. seznam(nestačí jen první/aktuální nastavit na
NULL)

void vypisSeznamPozic() // vypíše pozice se seznamu na každý řádek jednoho (používá opakovaně
funkci pro výpis jedné pozice)

void pridejPozice(stPozice * pozice) // přidá pozici na konec lineárního seznamu

stPozice *odeberPoziciZeSeznamu(int cisloPozice) //odebere pozici z lin. sez. jeho čísla, vrátí ji,
a následně dealokuje

stPozice *najdiPoziciZeSeznamu(int cisloPozice) //najde pozici z lin. sez. podle čísla

void zrusSeznamPozic() //dealokuje celý lin.sez.(nestačí jen první/akt nastavit na NULL)


void alokujPolePohovoru() //alokuje pole ukazatelů pro pohovory, výchozí dimenze=10, v případě
hrozícího přetečení se pole realokuje

void pridejPohovor(stPohovor * pohovor) //přidá pohovor do pole na

void zmenStavPohovor(int id, enum STAV_POHOVORU vysledek) //změní stav pohovoru

void vypisPohovory() //vypíše všechny pohovory

```

agenda.c

Obsahuje proměnné:

```

static stKandidat *prvniKandi = NULL, *aktKandi = NULL;
static stPozice *prvniPozice = NULL, *aktPozice = NULL;
static stPohovor **polePohovoru= NULL;

// static, aby byly vidět jen v tomto souboru
a implementuje funkce z agenda.h

```

Modul Main

Hlavní soubor *main.c* programu obsahuje logiku pro editaci pohovoru a dále menu pro volání funkcí modulu agenda.

Hlavní menu

1. Načti data do seznamu kandidátů/pozic ze souboru CSV
2. Vypiš seznam kandidátů/pozic
3. Přidej z klávesnice kandidáta/pozici do seznamu
4. Odeber kandidáta/pozici ze seznamu
5. Zruš seznam kandidátů/pozic
6. Najdi kandidáta/pozici
7. Přidej pohovor
8. Edituj stav pohovoru
9. Vypiš pohovory
10. Ukončit program – nutná dealokace všech alokovaných paměťových prostředků